

1 CLAIMS

2
3 1. A system comprising:

4 one or more computer-readable media;

5 an application program interface embodied on the one or more computer-
6 readable media and which can be utilized to present a presentation;

7 the application program interface comprising:

8 a plurality of *open* methods that can be called by an application, the
9 open methods comprising a parameter for specifying a destination for the
10 presentation, and at least a parameter for specify a data source for the
11 presentation; wherein collectively, the parameters for specifying a data
12 source enable data sources to be specified in different manners;

13 a method for ascertaining the state of a media engine that causes the
14 presentation to be presented;

15 a method for ascertaining capabilities of the media engine;

16 a method to start processing media samples that are the subject of
17 the presentation;

18 a method to stop processing media samples; and

19 a method to pause media sample processing.
20

21 2. The system of claim 1, wherein said one parameter for specifying a data
22 source specifies a URL as a data source.
23
24
25

1 3. The system of claim 1, wherein said one parameter for specifying a data
2 source specifies a media source created by the application.

3
4 4. The system of claim 1, wherein said one parameter for specifying a data
5 source specifies an object that has an interface from which a media source object
6 can be obtained.

7
8 5. The system of claim 1, wherein said one parameter for specifying a data
9 source specifies an object from which a byte stream can be obtained.

10
11 6. The system of claim 1, wherein said one parameter for specifying a data
12 source specifies a topology object.

13
14 7. The system of claim 1, wherein one media engine state comprises an initial
15 state that is assumed when the media engine is created.

16
17 8. The system of claim 1, wherein one media engine state comprises a
18 connecting state that is assumed when the media engine is trying to open a
19 networked media and is connecting to a server.

20
21 9. The system of claim 1, wherein one media engine state comprises an opened
22 state that is assumed when the media engine has successfully opened a media.

1 **10.** The system of claim 1, wherein one media engine state comprises a
2 running state that is assumed when the media engine has successfully started a
3 presentation.

4
5 **11.** The system of claim 1, wherein one media engine state comprises a paused
6 state that is assumed when the media engine has successfully paused a
7 presentation.

8
9 **12.** The system of claim 1, wherein one media engine state comprises a
10 shutdown state that is assumed when the media engine is shut down.

11
12 **13.** The system of claim 1, wherein one media engine state comprises a
13 transitioning state that is assumed when an asynchronous call is made on the
14 media engine.

15
16 **14.** The system of claim 1, wherein one media engine state comprises a
17 suspended state that is assumed when the opening or running of the media engine
18 is waiting on a user action.

19
20 **15.** The system of claim 1, wherein the method for ascertaining the capabilities
21 of the media engine returns a bitwise OR of the current capabilities of the media
22 engine.

1 **16.** The system of claim 1, wherein the method for ascertaining the capabilities
2 of the media engine returns a bitwise OR of the current capabilities of the media
3 engine, and wherein bits that describe the current capabilities of the media engine
4 comprise a start bit, a skipforward bit, a skip backward bit, a skip node bit, a seek
5 bit, and a pause bit.

6
7 **17.** The system of claim 1, wherein the method to start processing samples
8 comprises a first parameter for specifying a time format to use with other
9 parameters and at least a second parameter for specifying an offset from which to
10 start processing.

11
12 **18.** The system of claim 1 further comprising a method that returns a current
13 destination.

14
15 **19.** The system of claim 1 further comprising a method that returns a
16 presentation clock being used for the presentation.

17
18 **20.** The system of claim 1 further comprising a method that provides access to
19 metadata associated with the presentation.

20
21 **21.** The system of claim 1 further comprising a method that provides access to
22 statistics associated with the presentation.

1 **22.** The system of claim 1 further comprising an event generator interface for
2 generating events associated with the presentation.

3
4 **23.** The system of claim 1 further comprising an event generator interface for
5 generating events associated with the presentation, wherein one event comprises
6 an event that is associated with a new presentation.

7
8 **24.** The system of claim 1 further comprising an event generator interface for
9 generating events associated with the presentation, wherein one event comprises
10 an event associated with completion of an open call.

11
12 **25.** The system of claim 1 further comprising an event generator interface for
13 generating events associated with the presentation, wherein one event comprises
14 an event associated with completion of an operation begun by calling said method
15 to start processing media sample.

16
17 **26.** The system of claim 1 further comprising an event generator interface for
18 generating events associated with the presentation, wherein one event comprises
19 an event associated with completion of an operation begun by calling said method
20 to stop processing media samples.

1 **27.** The system of claim 1 further comprising an event generator interface for
2 generating events associated with the presentation, wherein one event comprises
3 an event associated with completion of an operation begun by calling said method
4 to pause processing media samples.

5
6 **28.** The system of claim 1 further comprising an event generator interface for
7 generating events associated with the presentation, wherein one event comprises
8 an event that indicates that a last media sample from an active media source has
9 been rendered.

10
11 **29.** The system of claim 1 further comprising an event generator interface for
12 generating events associated with the presentation, wherein one event comprises
13 an event associated with completion of an operation begun by calling a close
14 method on the media engine.

15
16 **30.** The system of claim 1 further comprising an event generator interface for
17 generating events associated with the presentation, wherein one event comprises
18 an event associated with a presentation switch.

19
20 **31.** The system of claim 1 further comprising an event generator interface for
21 generating events associated with the presentation, wherein one event comprises
22 an event associated with a destination change.

1 **32.** The system of claim 1 further comprising an event generator interface for
2 generating events associated with the presentation, wherein one event comprises
3 an event associated with a media engine state change.
4

5 **33.** The system of claim 1 further comprising an event generator interface for
6 generating events associated with the presentation, wherein one event comprises
7 an event that indicates that a set of operations allowed by the media engine has
8 changed.
9

10 **34.** The system of claim 1 further comprising a stream selector interface that
11 provides methods for setting stream selection modes.
12

13 **35.** The system of claim 1 further comprising a stream selector interface that
14 provides methods for setting stream selection modes, wherein one mode comprises
15 an automatic mode in which the media engine is responsible for selecting which
16 streams are used; another mode comprises a manual mode in which an application
17 has control over which streams are selected.
18

19 **36.** The system of claim 1 further comprising a media session interface that
20 provides methods that enable the media engine to configure a media session for a
21 presentation.
22
23
24
25

1 **37.** The system of claim 1 further comprising a media session interface that
2 provides methods that enable the media engine to configure a media session for a
3 presentation, wherein one method comprises a method for initializing a full
4 topology on the media session.

5
6 **38.** The system of claim 1 further comprising a media session interface that
7 provides methods that enable the media engine to configure a media session for a
8 presentation, wherein one method comprises a method for converting a partial
9 topology into a full topology.

10
11 **39.** The system of claim 1 further comprising a media session interface that
12 provides methods that enable the media engine to configure a media session for a
13 presentation, wherein one method comprises a method to start processing media
14 samples for the presentation.

15
16 **40.** The system of claim 1 further comprising a media session interface that
17 provides methods that enable the media engine to configure a media session for a
18 presentation, wherein one method comprises a method to start media sample
19 processing without rendering the media samples.

20
21 **41.** The system of claim 1 further comprising a media session interface that
22 provides methods that enable the media engine to configure a media session for a
23 presentation, wherein one method comprises a method to pause media sample
24 processing in the media session.

25

1 **42.** The system of claim 1 further comprising a media session interface that
2 provides methods that enable the media engine to configure a media session for a
3 presentation, wherein one method comprises a method to stop media sample
4 processing in the media session.

5
6 **43.** The system of claim 1 further comprising a media session interface that
7 provides methods that enable the media engine to configure a media session for a
8 presentation, wherein one method comprises a method to shut the media session
9 down and release resources used by the media session.

10
11 **44.** The system of claim 1 further comprising a media session interface that
12 provides methods that enable the media engine to configure a media session for a
13 presentation, wherein one method comprises a method to specify a presentation
14 clock to be used in rendering a current media session.

15
16 **45.** The system of claim 1 further comprising a media session interface that
17 provides methods that enable the media engine to configure a media session for a
18 presentation, wherein one method comprises a method to return a presentation
19 clock being used to rendering a current media session.

20
21 **46.** A system comprising:
22 one or more computer-readable media;
23 an application program interface embodied on the one or more computer-
24 readable media and which can be utilized to present a presentation;
25 the application program interface comprising:

1 a plurality of *open* methods that can be called by an application, the
2 open methods comprising a parameter for specifying a destination for the
3 presentation, and at least a parameter for specify a data source for the
4 presentation; collectively, the parameters for specifying a data source
5 enabling data sources to be specified in different manners;

6 a method for ascertaining the state of a media engine that causes the
7 presentation to be presented;

8 a method for ascertaining capabilities of the media engine;

9 a plurality of methods for providing presentation control;

10 a method that provides access to metadata associated with the
11 presentation;

12 a method that provides access to statistics associated with the
13 presentation;

14 an event generator interface for generating events associated with the
15 presentation;

16 a stream selector interface that provides methods for setting stream
17 selection modes; anda

18 a media session interface that provides methods that enable the
19 media engine to configure a media session for a presentation.
20
21
22
23
24
25